Beszédfelismerők mély neuronhálós állapotkapcsolási algoritmusainak kísérleti összehasonlítása

Tóth László[1], Grósz Tamás[1,2], Gosztolya Gábor[2]

[1]Szegedi Tudományegyetem, Informatikai Intézet
[2]MTA-SZTE Mesterséges Intelligencia Kutatócsoport
{ tothl, groszt, ggabor } @ inf.u-szeged.hu

Kivonat Az utóbbi években a rejtett Markov-modelles beszédfelismerőkben a kibocsátási eloszlások becslésére használt Gauss-keverékmodelleket a mély neuronhálók váltották fel. Azonban az elmúlt három évtizedben a Gauss-alapú modellezés javítására számos algoritmikai finomítást vezettek be, amelyeknek a mély neuronhálós környezetbe való átültetése nem mindig triviális. Egy ilyen példa a környezetfüggő beszédhangmodellek létrehozására kitalált döntési fa-alapú állapotkapcsolási eljárás. Jelenleg mindenki a régi, jól bevált algoritmust használja a mély hálós felismerőkben is, annak ellenére, hogy az algoritmus speciálisan a Gauss-görbék illeszkedését használja ki, így optimalitása egy mély hálós rendszerben megkérdőjelezhető. Az utóbbi időben azonban több olyan állapotklaszterező algoritmust is publikáltak, amelyek megkísérlik a korábbi eljárást a mély neuronhálós modellezéshez igazítani. Jelen cikkben négy ilyen algoritmust hasonlítunk össze egy angol nyelvű adatbázison, majd a legjobbnak bizonyuló módszert egy magyar korpuszon is kiértékeljük. Eredményeink azt mutatják, hogy ezek az új algoritmusok szignifikánsan jobb eredményt adnak, mint a régi, Gauss-alapú megoldás, köszönhetően annak, hogy működésük a neuronhálók kimenetéhez van igazítva.
Kulcsszavak: beszédfelismerés, mély neuronháló, környezetfüggő modellezés, állapotkapcsolás

1. Bevezetés

A gépi beszédfelismerésben használt rejtett Markov-modellek (Hidden Markov Model, HMM) legfontosabb komponense az akusztikus jellemzők eloszlását leíró valószínűségi modell, mely célra hagyományosan Gauss-eloszlások keverékét (Gaussian Mixture Model, GMM) szokták alkalmazni. Az utóbbi években azonban a mély neuronhálók (Deep Neural Network, DNN) annyival jobbnak bizonyultak ezen a téren, hogy a hagyományos, GMM-alapú rendszereket (röviden HMM/GMM) teljesen kiszorította a DNN-alapú hibrid megoldás (röviden HMM/DNN hibrid). A HMM/GMM modellek mögött azonban több évtizednyi fejlesztés van, számos olyan algoritmikai trükkel, amelyeket nem feltétlenül lehet a hibrid HMM/DNN sémára triviális módon átültetni. Egy ilyen példa a

beszédfelismerők betanítása részletes szegmentálási annotáció nélkül (angol szakzsargonban "flat start" tanítás), egy másik fontos feladat pedig a környezetfüggő (context-dependent, CD) beszédhang-modellek automatikus kialakítása. A hagyományos HMM/GMM keretben mindkettőre jól bevált algoritmusok léteznek, így továbbra is ezeket a módszereket használjuk a hibrid rendszerek inicializálása során. Ez gyakorlatilag azt jelenti, hogy a HMM/DNN modell betanítása egy HMM/GMM modell betanításával kezdődik, amelyet a fenti két részfeladat megoldása után eldobunk. Ez a módszer ugyan működőképes, de egyáltalán nem gazdaságos, és valószínűleg nem is optimális.

A flat start tanítás lényege a modellek iteratív újratanítása és újraillesztése, ami mögött a HMM/GMM rendszerek esetében jól kidolgozott matematikai háttér és komoly gyakorlati tapasztalat van. Ugyanezt mély hálós rendszerekkel megvalósítani nem magától értetődő, bár az utóbbi években többen megmutatták, hogy megfelelő odafigyeléssel nem lehetetlen [1,2,3,4]. A környezetfüggő modellezés mély hálós megvalósítása még kevésbé megoldott. A hibrid modellek feltalálói évtizedekig ragaszkodtak a környezetfüggetlen (context-independent, CI) modellezéshez [5], azóta azonban kiderült, hogy a környezetfüggő modellek a mély neuronhálós HMM/DNN hibridek pontosságát is szignifikáns mértékben növelik [6,7]. Ebből kifolyólag fontos lenne hatékony algoritmusokat találni a CD modellek automatikus kialakításához. Jelenleg mindenki a HMM/GMM rendszerek jól bevált döntési fa-alapú állapotklaszterező eljárását használja e célra [8]. Ez a módszer Gauss-eloszlásokat illeszt az egyes állapotokhoz tartozó adatpontokra, majd az illeszkedés pontossága alapján tart egyben vagy oszt szét egyes fonéma-állapotokat (a későbbi osztályokat). Ezt a lépést iteratívan ismételgetve az eljárás egy fa-jellegű hierarchikus klaszterezést eredményez. A módszer Gauss-alapú modellezés esetén remekül bevált, azonban erősen kétséges, hogy a Gauss-görbék illeszkedése mennyire jó kritérium, amikor a végső modell egy teljesen más reprezentáció, például egy neuronháló lesz.

További probléma, hogy a GMM- és a DNN-alapú rendszerek esetén az input reprezentálási módja is jelentősen eltérhet. HMM/GMM rendszerek esetében jól bevált az ún. mel-frekvenciás kepsztrális együtthatók (mel-frequency cepstral coefficient, MFCC) használata. Úgy tűnik azonban, hogy a HMM/DNN rendszerek némileg jobban működnek olyan primitívebb jellemzőkkel, mint például a mel-frekvenciasávok energiái [9]. A HMM/GMM modelleket azonban ezeken a jellemzőkön nem lehet jól betanítani. A jelenlegi módszer szerint tehát az MFCC jellemzőket is pusztán az állapotklaszterező eljárás kedvéért generáljuk le, majd utána eldobjuk. Ennél nyilván léteznie kell jobb megoldásnak is.

Az utóbbi néhány évben több szerző is megpróbálkozott a régi klaszterező eljárás DNN-ekhez való igazításával. A jellemzőkészletek eltéréséből eredő probléma elkerülésére azt a megoldást javasolták, hogy a klaszterezést a DNN kimenetén futtassuk le, ne pedig a bemenő jellemzőkön. Ezzel az egyszerű módszerrel többen is próbálkoztak [1,2,10,11]. Bár ez a huszárvágás kikerüli az eltérő jellemzőkészletek problémáját, a klaszterező eljárás ugyanaz marad, így a Gauss-os eloszlási feltevéstől továbbra sem szabadultunk meg.

Más szerzők a klaszterező eljárás döntési kritériumát is módosítják oly módon, hogy az jobban illeszkedjen a neuronhálós eloszlás-modellezéshez. Mivel a neuronháló kimenete egy diszkrét valószínűségi eloszlás közelítéseként értelmezhető, Gosztolya és tsai egy korábbi cikkükben döntési kritériumként a Kullback-Leibler divergencia használatát javasolták [12]. Módszerük alapötletét Imseng és tsai algoritmusa adta, amelyet eredetileg az ún. KL-HMM modellek betanításához találtak ki [13]. Zhu és tsai egy entrópia-alapú faépítő kritériumot javasoltak [14]. Végezetül, Wang és tsai egy speciális hálózaton alapuló klaszterezési megoldással álltak elő, amely mély kanonikus korrelációelemzésre (Deep Canonical Correlation Analysis) van optimalizálva [15].

A fenti szerzők mindegyike a felismerési szóhiba (word error rate, WER) csökkenését tapasztalta az új algoritmusok alkalmazásával a hagyományos GMM-alapú eljáráshoz képest. Azonban egyik szerzőcsoport sem hasonlította össze a cikkét a többi hasonló módszerrel. Ráadásul mindegyikük más-más adatbázist használt, ami az egyes módszerek egymással való összevetését teljesen lehetetlenné teszi. Jelen cikkben négy ilyen algoritmust hasonlítunk össze ugyanazon a nagyszótáras felismerési feladaton, ráadásul ugyanaz a neuronháló fogja szolgáltatni a klaszterező algoritmusok inputját. Megjegyezzük, hogy a kiindulási, környezetfüggetlen címkék illesztését a korábbiakban javasolt neuronhálós "flat-start" módszerrel végeztük [3], így ezen lépés során sem használtunk GMM modelleket. Ebből kifolyólag az összehasonlított rendszerek közül az a kettő, amelyikben a döntési kritérium DNN-ekhez van igazítva (azaz Gosztolya és tsai [12] valamint Zhu és tsai [14]) teljesen "GMM-mentes" megoldást ad a teljes tanítási folyamatra.

2. Döntési fa-alapú állapotkapcsolás

A döntési fa-alapú állapotkapcsolási algoritmus Young és tsai nevéhez fűződik [8], és mára a nagyszótáras beszédfelismerő rendszerek készítésének elengedhetetlen komponensévé vált. A módszer alapötlete, hogy az adott beszédhang (pontosabban modell-állapot) kontextusfüggő variánsait összevonjuk, majd a halmazt lépésről lépésre részhalmazokra osztjuk, ami egy hierarchikus, faként reprezentálható klaszterezést eredményez. Az egyes lépésekben az előre definiált kérdésekkel megadott felosztási lehetőségek közül az algoritmus mindig azt választja, amelyre az egyben hagyott, illetve a szétosztott halmazok maximálisan különböznek. Ennek a különbözőségnek, pontosabban a kiindulási $\mathcal{S}$ állapothalmaz szétosztásából származó nyereségnek a mérésére az algoritmus egy valószínűségi alapú döntési kritériumot használ. Ez a módszer annyira bevált, hogy az elmúlt húsz évben csak apró módosításokat javasoltak rajta (például a faépítést vezérlő, fonetikai ismeretet igénylő kérdések automatikus generálását [16]).

2.1. A valószínűségi (likelihood) alapú döntési kritérium

Tegyük fel, hogy adott állapotok egy $\mathcal{S}$ halmaza, amelyeket klaszterezni (majd ez alapján összekapcsolni) szeretnénk Young és tsai módszerével. A döntési fa építése során minden csomópontnál két nem átfedő részhalmazra osztjuk az aktuális

$\mathcal{S}$ halmazt az illeszkedő kérdések alapján, a kérdésre adott válasznak megfelelően. Az optimális szétosztás meghatározására Odell egy maximum likelihood alapú döntési kritériumot definiált [17], majd egy hatékony algoritmust is javasolt, amely a kritériumot az alábbi módon közelíti:

$$L(\mathcal{S}) \simeq -\frac{1}{2}\big(\log[(2\pi)^K|\Sigma(\mathcal{S})|] + K\big)\sum_{s\in\mathcal{S}} N(s), \tag{1}$$

ahol $s \in \mathcal{S}$ jelöli az egyes állapotokat, $\Sigma(\mathcal{S})$ az $\mathcal{S}$-be eső adatok szórását, $N(s)$ pedig az s halmazba eső példák (adatvektorok) számát. A fenti formula alapján minden lépésben azt a q kérdést fogjuk választani, amely maximizálja a likelihood érték $\Delta L(q|\mathcal{S})$ változását:

$$\Delta L(q|\mathcal{S}) = \big(L(\mathcal{S}_y(q)) + L(\mathcal{S}_n(q))\big) - L(\mathcal{S}), \tag{2}$$

ahol $\mathcal{S}_y(q)$ és $\mathcal{S}_n(q)$ a $\mathcal{S}$ halmaz két részhalmaza, amelyek a q kérdés alkalmazásával állnak elő. Megjegyezzük, hogy a likelihood értékek nem függnek direkt módon az egyes tanítópéldáktól, csupán az egyes állapotokra eső tanítópéldák szórásától, valamint a példák számától.

3. Neuronháló-alapú állapotkapcsolás

A fent ismertetett kritérium a példák szórására épül, ami Gauss-alapú modellezés esetén ésszerűen hangzik, hiszen a Gauss-görbék egyik fő paramétere a szórás. Ha azonban a GMM-alapú modellek helyett neuronhálóra épülő modelleket készítünk, akkor a fenti formula használata kérdésessé válik. Ehhez jön még a jellemzőkészletek esetleges eltéréséből fakadó korábban ismertetett dilemma.

E két problémára több szerző is azt a megoldást javasolta, hogy Gauss-görbék illesztése helyett tanítsunk be egy neuronhálót a kiindulási környezetfüggetlen címkék osztályozására (ez lehet hagyományos vagy mély háló is). Ezt a hálót a továbbiakban "segéd" neuronhálónak fogjuk nevezni, és röviden CI-NN jelöléssel fogunk rá hivatkozni. A betanításához szükséges CI címkék időbeli illesztését akár egy hagyományos HMM/GMM rendszer betanításával, akár az utóbbi időben javasolt DNN-alapú 'flat-start' technikák segítségével (pl. [2,3,4]) is elvégezhetjük. A lényeg jelen esetben, hogy az állapotkapcsoláshoz szükséges klaszterezést nem a bemenő jellemzőkön, hanem a segéd-neuronháló kimenetén végezzük el. Ily módon a klaszterezés során nem lesz jelentősége annak, hogy a segéd-neuronháló milyen jellemzőkön tanult. Ki kell azonban találnunk, hogy a klaszterező algoritmust miként igazítsuk a segédháló kimeneti értékeihez. A továbbiakban erre ismertetünk négy módszert.

3.1. A segéd-neuronháló kimenő vektorainak klaszterezése

Ez a Senior és tsai által javasolt megoldás a lehető legegyszerűbb. Ők közvetlenül a segédháló kimenetén futtatják le a klaszterző algoritmust, míg magát az

algoritmust semmilyen módon nem módosítják. A módszert ismertető cikkükben a szószintű hiba kis javulását tapasztalták [10]. Bár cikkük hangzatosan a "GMM-mentes" címet viseli, magán a Gauss-görbék illesztésén alapú klaszterezési módszeren semmit sem változtatnak, így véleményünk szerint a címadás legalábbis megtévesztő.

3.2. A DNN rejtett rétegének kimenetén végzett klaszterezés

Senior és tsai munkájával párhuzamosan Bacchiani és Rybach azt javasolta, hogy a klaszterezést a segédhálónak ne a kimenetén, hanem a legutolsó rejtett rétegén végezzük el [11]. Magán a klaszterezési eljáráson nem változtattak, ők is a jól bevált Young-féle algoritmust alkalmazták. Úgy találták, hogy kisebb állapotszámok esetén ez a megoldás jobban működött, mint a standard eljárás, habár nagy állapotszám esetén ez megfordult. Ezt a keretszintű címkék pontatlan illesztésével próbálták magyarázni, az illesztést ugyanis egy hagyományos HMM/GMM rendszerből vették át.

3.3. Kullback-Leibler divergencián alapuló klaszterezés

Az eddig ismeretett két módszer egyike sem változtatta meg a klaszterezés során használt döntési kritériumot, csupán az inputtal játszottak. Gosztolya és tsai viszont egy korábbi cikkükben javaslatot tettek egy neuronháló-kimenetekhez igazított döntési kritériumra [12], melynek alapötletét Imseng és tsai Kullback-Leibler HMM-ekhez kifejlesztett módszere adta [13]. Ez a módszer teljesen GMM-mentes, ahogy az alábbiakban röviden összefoglaljuk, [12], [18] és [19] alapján.

A neuronhálók kimenete egy diszkrét valószínűségi eloszlásként értelmezhető. Ilyen eloszlások összehasonlítására használhatjuk a Kullback-Leibler divergenciát, amely egy z_t és egy y_s vektor fölött így definiálható:

$$D_{KL}(y_s||z_t) = \sum_{k=1}^{K} y_s(k) \log \frac{y_s(k)}{z_t(k)}, \tag{3}$$

ahol $k \in \{1, \ldots, K\}$ a vektor dimenziója [20]. Ez alapján a likelihood érték maximalizálása helyett a KL-divergencia minimalizálására fogunk törekedni:

$$D_{KL}(\mathcal{S}) = \sum_{s \in S} \sum_{f \in F(s)} \sum_{k=1}^{K} y_{\mathcal{S}}(k) \log \frac{y_{\mathcal{S}}(k)}{z_f(k)}, \tag{4}$$

ahol $\mathcal{S}$ az s állapotok egy halmaza, $F(s)$ pedig az adott állapothoz tartozó tanítópéldák. Az $\mathcal{S}$-hez tartozó példákat reprezentáló vektor (azaz $y_{\mathcal{S}}$) úgy kapható meg, mint az $\mathcal{S}$-be eső példák mértani közepe, bár Imseng eredeti cikkében megemlíti, hogy a számtani közép is alkalmazható e célra [21]. Kibontás és egyszerűsítés után azt kapjuk, hogy

$$D_{KL}(\mathcal{S}) = -\sum_{s \in \mathcal{S}} N(s) \log \sum_{k=1}^{K} y_{\mathcal{S}}(k), \tag{5}$$

vagyis az állapotok egy $\mathcal{S}$ halmazának KL-divergenciája az egyes állapotokhoz tartozó y_s és $N(s)$ statisztikai értékekből kapható meg [18]. A $\mathcal{S}$ állapothalmaz szétosztásakor magától értetődően azt a kérdést érdemes választani, amelyre a KL-divergencia $\Delta D_{KL}(q|\mathcal{S})$ változása maximális:

$$\Delta D_{KL}(q|\mathcal{S}) = D_{KL}(\mathcal{S}) - \big(D_{KL}(\mathcal{S}_y(q)) + D_{KL}(\mathcal{S}_n(q))\big). \tag{6}$$

3.4. Entrópián alapuló állapotklaszterezés

Zhu és tsai egy negyedik megoldási lehetőséget javasoltak [14]. Ez szintén az Odell-féle kritérium (Eq. (1)) egy Gauss-görbék illeszkedésétől független formulával való helyettesítésén alapszik, konkrétan az összevont klaszterekbe tartozó példák hasonlóságát az entrópiájuk alapján formalizálja. A példavektorok alapján egy K-dimenziós eloszlás entrópiája az alábbi módon becsülhető:

$$H(p) = \sum_{i=1}^{K} p(i) \log p(i). \tag{7}$$

A kiinduló állapotokat leíró eloszlások (azaz az y_s vektorok) becslésére pedig az adott állapotokhoz tartozó neuronháló-kimenetek átlagát fogjuk használni. Továbbmenve, állapotok egy $\mathcal{S}$ halmaza esetén a prototípus valószínűségi vektort ($y_{\mathcal{S}}$) a halmazba eső állapotok y_s vektorainak súlyozott átlagaként állítjuk elő, ahol a súlyozás a $N(s)$ példaszámok alapján történik. Maga a döntési kritérium pedig az entrópiafüggvény alapján formalizálható az alábbi módon:

$$D_E(\mathcal{S}) = -\sum_{k=1}^{K} N(\mathcal{S}) y_{\mathcal{S}}(k) \log y_{\mathcal{S}}(k). \tag{8}$$

4. DNN Flat Start

Az állapotklaszterezés előtt el kell végeznünk a CI címkék időbeli illesztését, majd a címkék osztályozására be kell tanítanunk a segéd-neuronhálót. Az illesztést elvégezhetjük hagyományos módon, egy HMM/GMM rendszer betanításával is. Bacchiani and Rybach azonban arra figyelmeztetnek, hogy a kapott illesztés nem feltétlenül lesz optimális a DNN tanítása céljára [11]. Ebből kifolyólag mi egy elegánsabb megoldást választottunk, amely nem igényli GMM-alapú modellek betanítását. A CI-NN neuronhálónkat 'flat start' módon tanítottuk be a Maximum Mutual Information (MMI) tanítási kritérium segítségével, Gosztolya és tsai korábbi cikkét követve [3]. A hatékonyság növelésére a tradicionális MMI tanítási eljáráson az alábbi módosításokat végeztük. Elsőként, a tanítási célvektorokat a HMM-ek forward-backward algoritmusával állítottuk elő. Másodsorban, amíg a szokványos eljárás az MMI-tanítást csak a tanítási folyamat végén, szószintű címkékkel végzi (lásd pl. [22,23]), mi puszta fonetikai címkesorral dolgoztunk, nyelvi modell nélkül. Ezeknek az egyszerűsítéseknek köszönhetően

a fonetikai címkék újraillesztése minden mondat (input fájl) után elvégezhető, gyors konvergenciát eredményezve. Ezzel a módszerrel a DNN tanítása nem csupán gyorsabb lett más módszerekhez képest (pl. [2]) hanem némileg jobb eredményt is adott.

5. Kísérleti beállítások

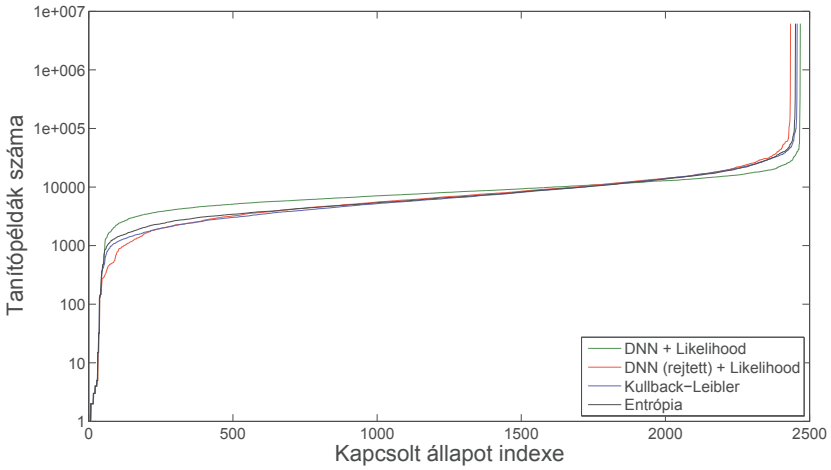
Mély neuronhálón alapuló felismerőnket a 81 órányi angol olvasott szöveget tartalmazó Wall Street Journal (WSJ) korpusz `si-284` jelű részhalmazán [24] tanítottuk be. A kiértékelést az `eval92` és `eval93` elnevezésű teszthalmazokon végeztük, az ún. „open-vocabulary" (azaz hatvanezer szót tartalmazó) szótárral, a szokványos szűkített trigram nyelvi modellel. Ezek közül fejlesztői (development) halmazként az `eval93` elnevezésű rész-korpusz szolgált, azaz ezen optimalizáltuk az olyan meta-paramétereket, mint a nyelvi modell súlya, a szó-beszúrási büntetés, illetve a kapcsolt állapotok száma. A tesztelés az ily módon optimalizált paraméterértékek mellett az `eval92` nevű részhalmazon történt.

A modellezéshez a saját belső fejlesztésű neuronhálós csomagunkat használtuk. Input jellemzővektorként 40 mel-skálás energiaegyüttható szolgált, az ezekhez tartozó első és második derivált-jellegű jellemzőkkel kiegészítve. A neuronháló bemenetét egy 15 szomszédos jellemzővektorból álló adatblokk képezte. Mind a CI segédháló, mind a végleges CD neuronháló öt rejtett réteget tartalmazott, rétegenként 1000-1000 neuronnal, az ún. ReLU aktivációs függvényt használva [25]. A kimenő rétegben a szokványos módon szoftmax aktivációt használtunk. A dekódolás és kiértékelés a HTK programcsomag neuronhálókhoz igazított változatával történt [26]. A klaszterező algoritmus négy változatát négy-négy paraméterbeállítás mellett futtattuk le, ezekben a klaszterezést vezérlő küszöböt úgy igyekeztünk belőni, hogy körülbelül 1800, 2400, 3000, illetve 3600 kapcsolt állapotot kapjunk.

A 1. ábra mutatja az állapotkapcsolás után előállt címkék darabszámának eloszlását a közelítőleg 2400 állapotot eredményező beállítások mellett. Amellett, hogy az eloszlás minden esetben nagyon hasonló, azt is megfigyelhetjük, hogy a leginkább kiegyensúlyozott eloszlást a Senior-féle megoldás (azaz a klasszikus klaszterező algoritmus DNN-kimeneten való lefuttatása) adta.

6. Eredmények és diszkusszió

Az 1. táblázat mutatja a development halmazon elért legjobb szóhiba-értékeket az egyes algoritmusokkal, valamint az ugyanazon paraméter-beállításokkal a teszthalmazon kapott hibát. Láthatóan a legegyszerűbb algoritmus működött a legroszabban, azaz a CI-DNN segédháló outputjain a klasszikus klaszterező algoritmust futtatva 8,7%-os szóhibát értünk el a development halmazon, míg a teszthalmazon 6,47%-ot. Ugyanazt a standard klaszterező algoritmust a háló legutolsó rejtett rétegének kimenetén futtatva (Bacchiani és Rybach cikkét követve) a development halmazon ugyan némileg nagyobb hibát kaptunk, a teszthalmazon viszont szignifikánsan javult az eredmény az előző módszerhez képest.



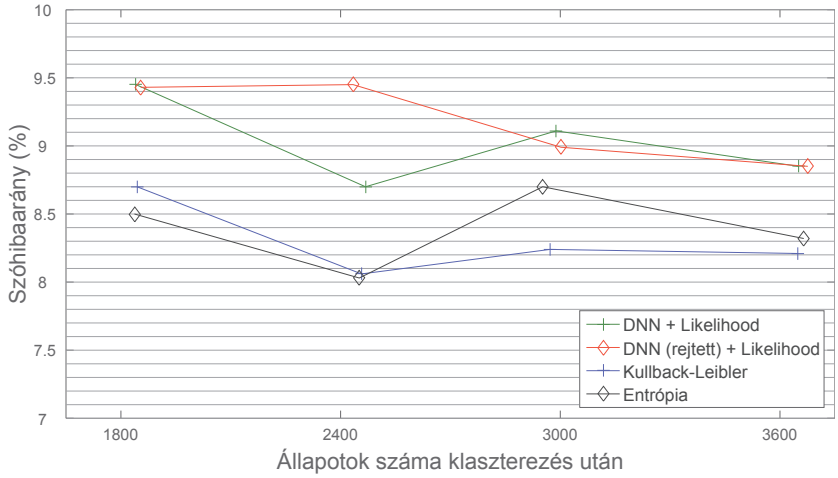
1. ábra. A tanító adatvektorok számának eloszlása a különböző klaszterező eljárásokkal kapott címkék esetén, kb. 2400 kapcsolt állapot mellett.

Klaszterezés inputja (DNN réteg)	Klaszterezés döntési kritériuma	WER	
		Dev.	Teszt
Kimeneti	Likelihood (Senior, [10])	8,70%	6,47%
Rejtett	Likelihood (Bacchiani, [11])	8,85%	6,04%
Kimeneti	KL-divergencia (Gosztolya, [12])	8,06%	5,72%
	Entrópia (Zhu, [14])	8,03%	5,92%

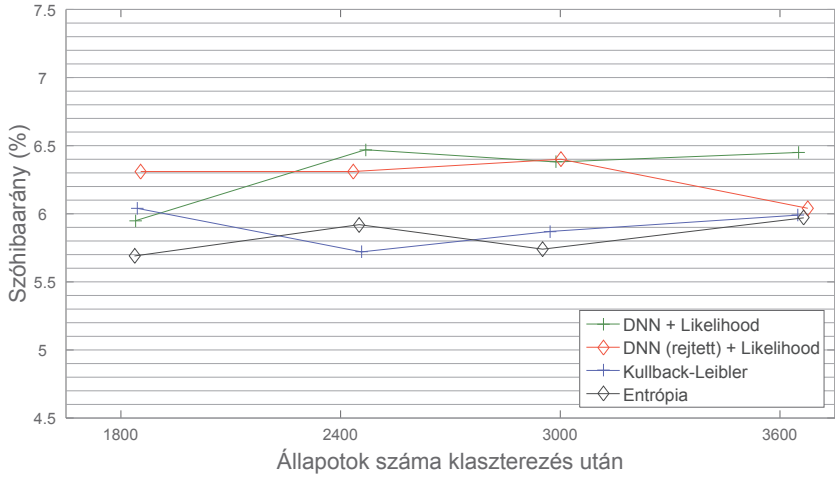
1. táblázat. A development halmazon elért legjobb szóhiba-arányok (WER) összegzése a hozzájuk tartozó teszthalmazon elért hibával, a különféle állapotkapcsoló algoritmusok esetén.

A további két tesztelt módszer nem a standard, Gauss-alapú klaszterezést alkalmazta, hanem egy új, DNN-ekhez igazított döntési kritériumot, és ez látszik is a hatékonyságukon. A development halmazon a két módszer gyakorlatilag megegyező szóhiba-arányt ért el (8,06% a Kullback-Leibler, illetve 8,03% az entrópia-alapú kritérium esetén); a teszhalmazon kicsit nagyobb különbséget mértünk, de ez még mindig nem statisztikailag szignifikáns. Összességében a Kullback-Leibler távolságon alapuló döntési kritériummal 0,8% abszolút hibacsökenést értünk el a Senior-féle triviális módszerhez képest, ami relatív skálán 12% hibacsökkenésnek felel meg.

A 2. és a 3. ábra mutatja a részletes eredményeket, azaz a négy algoritmus által a különböző állapotszámok esetére adott szóhiba-arányokat. Az ábrán szembetűnő módon szétválik a két standard hibakritériumot, valamit a két DNN-ekhez igazított hibakritériumot alkalmazó módszer, mivel az utóbbi csoport állapotszámtól függetlenül konzisztensen kisebb hibaértékeket ért el. Álláspontunk



2. ábra. A négy klaszterező algoritmus szóhibaaránya a development halmazon.



3. ábra. A négy klaszterező algoritmus szóhibaaránya a teszthalmazon.

szerint ez az eredmény nyilvánvaló módon azt tükrözi, hogy az állapotklaszterező algoritmus inputjának DNN-ekhez való igazítása mellett magát az algoritmust vezérlő döntési kritériumot is érdemes a neuronhálókhoz szabni, mivel így meggyőzően jobb eredményeket kaphatunk.

Az ábrákon látható görbék alapján a kapcsolt állapotok számának növelése javítja a klasszikus, likelihood-alapú klaszterezést használó módszerek teljesítményét. A két módosított kritériumot alkalmazó algoritmus esetén azonban az optimumpont valahol 2400 állapot környékén van a development halmazon. A teszthalmazon ezzel szemben nem lehet egyértelmű tendenciákat megfigyelni, mind a négy módszer relatíve érzéketlen az állapotszám növelésére. E téren

Klaszterezési algoritmus	WER	
	Dev.	Teszt
GMM + Likelihood	17,8%	17,3%
DNN + KL-divergencia	16,3%	15,9%
DNN + Entrópia	17,2%	16,3%

2. táblázat. A két legjobb módszer eredményei a magyar nyelvű korpuszon

meggyőző tanulságok levonásához valószínűleg még egy nagyságrenddel több tanítóadat lenne szükséges. Végezetül megjegyezzük, hogy a kapott optimális állapotszámok kisebbek, mint amekkorával a SWJ korpuszon rutinszerűen dolgozni szoktak. Ezért már magában a kisebb számításigény miatt is érdemesebb az általunk javasolt algoritmust használni.

7. Magyar nyelvű eredmények

A korábbiakban ismertetett kiértékelést az angol nyelvű Wall Street Journal beszédkorpuszon végeztük, egyrészt a nemzetközi publikálhatóság miatt, másrészt mivel nagyobb méretű, mint a rendelkezésünkre álló magyar adatbázisok. Magyar szerzőként azonban a magyar nyelvű beszédfelismerés fejlesztésében vagyunk érdekeltek, így a két legjobban teljesítő modell összevetését magyar nyelvű korpuszon is elvégeztük. E kísérletben a "Szeged" fantázianevű, 28 órányi híradófelvételt tartalmazó korpuszt használtuk. Mivel mind a korpusz, mind a modellezési paraméterek tökéletesen megegyeztek a korábbi publikációkban használatakkal, így az adatok és a rendszer részletes ismertetéstől eltekintük (megtalálható pl. [27]-ben).

A 2. táblázat mutatja a magyar nyelvű adatábázison kapott legjobb eredményeket a két módosított kritériumot alkalmazó (azaz a KL-divergencián, illetve az entrópián alapuló) algoritmus esetében. Viszonyítási pontként a GMM-alapon elvégzett címkeillesztést és állapotklaszterezést alkalmazó algoritmust tüntettük fel. Az eredményekből az látszik, hogy a KL-divergencián alapuló megoldás mind a development, mind a teszthalmazon némileg jobban teljesített, mint az entrópia-alapú kritérium, a relatív hibacsökkenés pedig a GMM-alapú technológiához képest kb. 8%.

8. Összegzés

Jelen cikkben négy, környezetfüggő állapokészlet kialakítására szolgáló klaszterező módszer teljesítményét hasonlítottuk össze mély neuronhálós környezetben. A négy algoritmus mindegyike egy segéd-neuronháló kimenetén (illetve rejtett rétegének kimenetén) végezte a klaszterezést. Kísérleteink azonban azt mutatták, hogy a klaszterezés bemenő jellemzői mellett érdemes az algoritmus döntési kritériumát is a neuronháló használatához igazítani, a módosított kritériummal

dolgozó algoritmus-változatok ugyanis minden esetben jobban teljesítettek, mint a hagyományos képletet használók. A kiértékelést az angol nyelvű Wall Street Journal beszédkorpuszon végeztük, de a legjobban teljesítő két módszert egy magyar nyelvű híradós beszédadatbázison is összevetettük. Mivel a kísérletek során a korábban javasolt MMI-alapú 'flat-start' tanítási módszert használtuk az inicializáláshoz, a Kullback-Leibler divergencián alapú megoldásunk sem az címkék illesztése, sem a környezetfüggő címkék kialakítása során nem igényli Gauss-görbék, azaz a "régi technológia" alkalmazását (a teljes GMM-mentes illesztési megoldás részletes ismertetése megtalálható [27]-ban).

Köszönetnyilvánítás

Tóth László munkáját az MTA Bolyai János Kutatási Ösztöndíja támogatta. Grósz Tamást az Emberi Erőforrások Minisztériuma UNKP-17-3 kódszámú Új Nemzeti Kiválóság Programja támogatta. A cikk elkészítéséhez használt Titan-X grafikus kártyát az NVIDIA Corporation adományozta.

Hivatkozások

1. Senior, A., Heigold, G., Bacchiani, M., Liao, H.: GMM-free DNN training. In: ICASSP. (2014)
2. Zhang, C., Woodland, P.: Standalone training of context-dependent Deep Neural Network acoustic models. In: ICASSP. (2014) 5597–5601
3. Gosztolya, G., Grósz, T., Tóth, L.: GMM-free flat start sequence-discriminative DNN training. In: Interspeech, San Francisco, USA (2016) 3409–3413
4. Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S.: Purely sequence-trained neural networks for ASR based on lattice-free MMI. In: Interspeech, San Francisco, USA (2016) 2751–2755
5. Bourlard, H., Morgan, N.: Connectionist Speech Recognition – A Hybrid Approach. Kluwer Academic (1994)
6. Yu, D., Deng, L., Dahl, G.: Roles of pretraining and fine-tuning in context-dependent DNN-HMMs for real-world speech recognition. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning. (2010)
7. Dahl, G., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained Deep Neural Networks for large vocabulary speech recognition. IEEE Trans. ASLP **20**(1) (2012) 30–42
8. Young, S.J., Odell, J.J., Woodland, P.C.: Tree-based state tying for high accuracy acoustic modelling. In: HLT. (1994) 307–312
9. Mohamed, A., Dahl, G.E., Hinton, G.: Acoustic modeling using Deep Belief Networks. IEEE Trans. ASLP **20**(1) (2012) 14–22
10. Senior, A., Heigold, G., Bacchiani, M., Liao, H.: GMM-free DNN acoustic model training. In: ICASSP. (2014) 5639–5643
11. Bacchiani, M., Rybach, D.: Context dependent state tying for speech recognition using deep neural network acoustic models. In: ICASSP. (2014) 230–234
12. Gosztolya, G., Grósz, T., Tóth, L., Imseng, D.: Building context-dependent DNN acousitc models using Kullback-Leibler divergence-based state tying. In: ICASSP, Brisbane, Ausztrália (2015) 4570–4574

13. Razavi, M., Rasipuram, R., Magimai-Doss, M.: On modeling context-dependent clustered states: Comparing HMM/GMM, hybrid HMM/ANN and KL-HMM approaches. In: ICASSP. (2014)
14. Zhu, L., Kilgour, K., Stüker, S., Waibel, A.: Gaussian free cluster tree construction using Deep Neural Network. In: Interspeech, Drezda, Németország (2015) 3254–3258
15. Wang, W., Tang, H., Livescu, K.: Triphone state-tying via Deep Canonical Correlation Analysis. In: Interspeech, San Francisco, USA (2016) 3444–3448
16. Beulen, K., Ney, H.: Automatic question generation for decision tree based state tying. In: ICASSP. (1998) 805–808
17. Odell, J.: The Use of Context in Large Vocabulary Speech Recognition. PhD thesis, University of Cambridge (1995)
18. Imseng, D., Dines, J.: Decision tree clustering for KL-HMM. Technical Report Idiap-Com-01-2012, Idiap Research Institute (2012)
19. Imseng, D., Dines, J., Motlicek, P., Garner, P., Bourlard, H.: Comparing different acoustic modeling techniques for multilingual boosting. In: Interspeech. (2012)
20. Kullback, S., Leibler, R.: On information and sufficiency. Ann. Math. Statist. **22**(1) (1951) 79–86
21. Imseng, D.: Multilingual speech recognition A posterior based approach. PhD thesis, École Polytechnique Fédérale de Lausanne (2013)
22. Kingsbury, B.: Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling. In: ICASSP. (2009) 3761–3764
23. Veselý, K., Ghoshal, A., Burget, L., Povey, D.: Sequence-discriminative training of deep neural networks. In: Interspeech. (2013) 2345–2349
24. Paul, D.B., Baker, J.M.: The design for the Wall Street Journal-based CSR corpus. In: HLT, Stroudsburg, PA, USA, Association for Computational Linguistics (1992) 357–362
25. Tóth, L.: Phone recognition with deep sparse rectifier neural networks. In: ICASSP. (2013) 6985–6989
26. Young, S., Evermann, G., Gales, M.J.F., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P.: The HTK Book. Cambridge University Engineering Department, Cambridge, UK (2006)
27. Grósz, T., Gosztolya, G., Tóth, L.: Mély neuronhálós beszédfelismerők GMM-mentes tanítása. In: MSZNY, Szeged (2017) 170–180